UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/745,020	12/20/2000	Christopher B. Wilkerson	884.370US1	8132

7590 09/08/2004

Schwegman, Lundberg, Woessner & Kluth, P.A.
P.O. Box 2938
Minneapolis, MN 55402

EXAMINER	
INOA, MIDYS	

ART UNIT	PAPER NUMBER
2188	

DATE MAILED: 09/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/745,020	WILKERSON, CHRISTOPHER B.	
	Examiner	Art Unit	
	Midys Inoa	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 July 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,6,7,9,10,12 and 14-31* is/are rejected.

7)☒ Claim(s) *4,5,8,11 and 13* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 July 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. The replacement drawings were received on July 19th, 2004. These drawings are accepted by the examiner.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 12, 14-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Improving Data Cache Performance by Pre-executing Instructions Under a Cache Miss" by James Dundas and Trevor Mudge (Dundas et al.) in view of Ukai et al. (5,983,324).

Regarding Claims 1, 12 and 29, Dundas et al. teaches a technique for improving data cache performance by pre-executing future instructions under a data cache miss ("run-ahead execution", Abstract). In this system, the normal mode is that of executing current instructions ("valid instructions") and the run-ahead mode is that of pre-executing future instructions. Since the pre-execution occurs under a cache miss, **a cache miss is an indication of a "run-ahead execution".** Dundas et al. does not teach setting a protection bit associated with the data that was pre-executed, thus protecting it from eviction prior to its use. Although Dundas discloses placing cache misses generated during run-ahead in a data memory access queue, **this does not imply that the cache misses in this queue will not be dealt with during run-ahed**; it only states that the misses are being placed in a queue to be dealt with in a certain order. Once the

cache misses are executed in the queued order and pre-fetches are being caused by such cache misses, **eviction becomes necessary in the system of Dundas et al**. Ukai et al. teaches a prefetch method which suppresses replacement of prefetched data (through means of protection settings) by other data before the prefetched data is used (Column 2, lines 47-67). The protection of such data ceases once the data has been accessed or used ("clearing protection bits during execution of normal threads"). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the prefetch method of Ukai et al. with the Pre-execution method of Dundas et al. since **both methods are involved with obtaining data prior to being needed** and integrating the method of Ukai to Dundas' Pre-execution method would prevent Pre-executed data from being lost before it is put to use by the system. Additionally, Dundas et al. does not teach executing prefetch and other programs in different threads. Ukai discloses processes running at the same time ("Multithreading", Column 1, lines 44-59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the multiple threads of Ukai et al. with the system of Dundas et al. since doing so would allow the system to execute more than one process at a time, thus, making it more efficient.

Regarding Claims 14-15, it is known in the art that prefetching algorithms are implemented reliably in software (as mentioned by Dundas, page 68, Column 2), therefore, software implementation instructions for a prefetch come from a specific section in a program. In addition, since software programs need to be compiled prior to their execution, it is understood that a prefetching algorithm may too come from a compiler.

Regarding Claims 16, 21, and 25, Dundas et al. teaches a technique for improving data cache performance by pre-executing future instructions under a data cache miss ("run-ahead

execution", Abstract) which is independent from normal execution of the system. In this system, the normal mode is that of executing current instructions ("valid instructions") and the run-ahead mode is that of pre-executing future instructions, which are deemed invalid because since they are future instructions no valid results can be produced at the present time. The system of Dundas et al. must include a cache; register files (holding instructions); and prefetching circuitry to perform prefetches when a cache miss occurs (see Abstract). Dundas et al. does not teach setting a protection bit associated with the data that was pre-executed, thus protecting it from eviction prior to its use. Although Dundas discloses placing cache misses generated during run-ahead in a data memory access queue, **this does not imply that the cache misses in this queue will not be dealt with**; it only states that the misses are being placed in a queue to be dealt with in a certain order. Once the cache misses are executed in the queued order and pre-fetches are being caused by such cache misses, **eviction becomes necessary in the system of Dundas et al**. Ukai et al. teaches a prefetch method employed in a cache system where a cache area with a plurality of lines may load target data ("data to be executed") to an user buffer in the processor ("plurality of registers") in order for the processor to access it (Figure 1). The prefetch method of Ukai suppresses replacement of prefetched data in the cache (through means of protection settings) by other data before the prefetched data is used (Column 2, lines 47-67). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the prefetch method of Ukai et al. with the Pre-execution method of Dundas et al. since **both methods are involved with obtaining data prior to being needed** and integrating the method of Ukai to Dundas' Pre-execution method would prevent Pre-executed data from being lost before it is put to use by the system. Additionally, it would have

been obvious to one of ordinary skill in the art at the time the invention was made to modify system of Dunadas' in view of Ukai to include more than one cache and more than one processor, so that each cache is related to one processor and so that the pre-execution and protection method could still be applied when more than one processor is present, thus making the methods of this system more versatile and universal to single processor and multiprocessor systems.

Regarding Claims 17 and 18, the protection mechanism disclosed by Ukai et al. includes identifiers in the form of an EMPTY free list and AGE free list which help in determining which of the prefetched data (or pre-executed data) that has yet to be used by the processor, thus indicating which of the prefetched data remains protected (Column 15). Additionally, Ukai discloses a cache entry header in which a use expectation flag indicates if such block of data has been used by the processor; use by the processor indicates if there is a need to further protect the data in question.

Regarding Claims 19 and 24, Ukai et al disclose a cache entry header (cache directory) holding detailed metadata information such as hit/miss information (in the form of a prefetch flag), and protection information in the form of a use expectation flag, where the expectation flag indicates if the data has been used by the processor, in which case, when used, protection is no longer necessary (see Figure 5).

Regarding Claims 26-28, the Pre-execution and protection method of Dundas in view of Ukai is described as to work for any cache in any computer system where pre-execution occurs. Therefore, such method could be adapted for use in an L1 cache, an L2 cache or an on die cache.

4. Claims 2-3, 6-7, 9-10, and 30-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dundas et al. in view of Ukai et al. and further in view of Petrick et al. (5,920,889).

Regarding Claims 2-3 and 30-31, the combination of Dundas et al. in view of Ukai discloses the invention as set forth by Claim 1 above. The combination of Dundas et al. in view of Ukai et al. does not teach evicting a cache line in order to store the pre-executed (or prefetched) data into the cache. Petrick et al. teaches a prefetch method in which a cache line is evicted from the cache in order to make room for prefetched data (Column 5, lines 46-63). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the eviction policy of Petrick et al. with the invention portrayed by the combination of Dundas et al. and Ukai et al. since adding this policy would ensure that the cache always has room to receive the pre-executed data by evicting unnecessary data. In performing this eviction policy within the invention of Dundas et al. in view of Ukai et al. the Pre-execution method would evict a line from the cache, store the Pre-executed data in such cache line, and protect it from being prematurely evicted.

Regarding Claims 6, 7, 9, and 10, Dundas et al. teaches a technique for improving data cache performance by pre-executing future instructions under a data cache miss ("run-ahead execution", Abstract). Dundas et al. does not teach setting a protection bit associated with the data that was pre-executed, thus protecting it from eviction prior to its use. Ukai et al. teaches a prefetch method which suppresses replacement of prefetched data (through means of protection settings) by other data before the prefetched data is used (Column 2, lines 47-67). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the

prefetch method of Ukai et al. with the Pre-execution method of Dundas et al. since **both methods are involved with obtaining data prior to being needed** and integrating the method of Ukai to Dundas' Pre-execution method would prevent Pre-executed data from being lost before it is put to use by the system. The combination of Dundas et al. in view of Ukai et al. does not teach evicting a cache line in order to store the pre-executed (or prefetched) data into the cache. Petrick et al. teaches a prefetch method in which a cache line is evicted from the cache in order to make room for prefetched data (Column 5, lines 46-63). It would have been obvious to one of ordinary skill in the art at the time the invention was made to integrate the eviction policy of Petrick et al. with the invention portrayed by the combination of Dundas et al. and Ukai et al. since adding this policy would ensure that the cache always has room to receive the pre-executed data. In performing this eviction policy within the invention of Dundas et al. in view of Ukai et al. the Pre-execution method would evict a line from the cache ("victim"), store the Pre-executed data in such cache line, and protect it from being prematurely evicted.

5. Claims 20, 22, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dundas et al. in view of Ukai et al. and Microsoft Computer Dicitionary, where the computer dictionary is used as an evidentiary reference.

Dundas et al in view of Ukai teach the method of the invention. They do not disclose a cache controller implementing cache strategies for moving data and using a cache directory and identifiers. Microsoft Computer Dicitionary discloses a controller used to control access in a computer system (Page 111). Caches are known to have controllers to aid in the movement and management of data by using resources such as cache directories and identifiers. The caches of Dundas must have cache controllers to aid with the method of the invention.

Allowable Subject Matter

6. Claims 4-5, 8, 11, and 13 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The Prior art of record does not teach a run-ahead cache system using protection bits to prevent pre-executed data from being evicted, and clearing the protection bits when there is a change in execution (i.e. when execution changes from normal to run-ahead or from run-ahead to normal).

Response to Arguments

7. Applicant's arguments filed on January 12th, 2004 regarding claims 1-31 have been fully considered but they are not persuasive.

8. In response to Applicant's argument that there is no suggestion to combine the references, the Examiner recognizes that references cannot be arbitrarily combined and that there must be some reason why one skilled in the art would be motivated to make the proposed combination of primary and secondary references. In re Nomiya, 184 USPQ 607 (CCPA 1975). However, there is no requirement that a motivation to make the modification be expressly articulated. The test for combining references is what the combination of disclosures taken as a whole would suggest to one of ordinary skill in the art. In re McLaughlin, 170 USPQ 209 (CCPA 1971), references are evaluated by what they suggest to one versed in the art, rather than by their specific disclosures. In re Bozek, 163 USPQ 545 (CCPA) 1969.

Applicant's argument is based on assuming that the invention of Dundas does not intend to execute cache misses during run-ahead when it places these missed in a queue. This is

incorrect. Placing such misses in a queue does not imply that such misses will only be dealt with after run-ahead, the system can in fact be dealing with the misses in the queue's order while run-ahead execution is taking place. In such a case, protection of the pre-executed data is necessary and thus, a need for the modification of the system to include the limitations of Ukai can be understood.

In this instance, Dundas et al. discloses the pre-execution method, Ukai et al. discloses a method of protecting prefetched data, and Petrick discloses a method of evicting data from a cache in order to make room for newly prefetched data. In combining these references, the method of Ukai can be applied to the pre-execution data of Dundas thus allowing the system to protect the pre-executed data; additionally, the method Petrick can be applied to the combination of Dundas and Ukai in order to allow the system to make room for more pre-execution data by evicting previously pre-executed data that no longer needs to be kept. Additionally, although Dundas does not specifically recognize "loosing pre-executed data prior to its use" or "running out of room in the cache for more pre-executed data" these are issues that are well known in the art when it comes to caching systems.

9. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

10. With respect to the independent claims, Dundas does not teach cache line protection; this limitation is being taught by Ukai. Ukai does not teach a normal mode during which instructions are executed directly from a cache, and a run-ahead mode where future instructions are being executed; these limitations are being taught by Dundas.

Additionally, Applicant argues that adding the reference of Petrick et al. to the invention of Dundas in view of Ukai creates the problem being solved by the application, where the problem was not previously present in the combination. This statement assumes that the invention of Dundas does not intend to execute cache misses during run-ahead when it places these missed in a queue. This is incorrect. Placing such misses in a queue does not imply that such misses will only be dealt with after run-ahead, the system can in fact be dealing with the misses in the queue's order while run-ahead execution is taking place. In such a case, protection of the pre-executed data is necessary and thus, a need for the modification of the system to include the limitations of Ukai can be understood. This in it of it's self hints at the need to evict data to make room for the prefetches that arise from the cache misses during run-ahead execution; thus, the teachings of Petrick et al. would be beneficial to the system of Dundas in view of Ukai.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Midys Inoa whose telephone number is (571) 272-4207. The examiner can normally be reached on M-F 7:00am - 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mano Padmanabhan can be reached on (571) 272-4210. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Midys Inoa
Examiner
Art Unit 2188

MI

9/7/04

MANO PADMANABHAN
SUPERVISORY PATENT EXAMINER